一种结合主题模型的推荐算法

曹占伟，胡晓鹏

(西南交通大学 信息科学与技术学院, 成都 611756)

摘 要：针对传统协同过滤推荐算法存在的冷启动、数据稀疏以及相似度度量的准确性问题，基于 LDA 主题模型对文本隐式主题挖掘的有效性和 KL 散度在主题分布相似性度量的准确性，提出了结合 LDA 主题模型的矩阵分解推荐算法。首先，利用改进的 LDA 算法输出项目-主题分布，并用困惑度作为主题数设置的修正函数；然后分别基于余弦相似度和 KL 散度计算得到项目相似度矩阵，将得到的相似度矩阵结合原评分训练集输出预评分，再将预评分填充到训练集；最后将训练集输入 ALS 矩阵分解算法得到推荐结果。通过 MovieLens 数据集的实验结果表明，该算法在不同隐式参数设定下均能得到比 ALS 推荐算法以及更小的预测误差，并且最优预测误差小于传统推荐算法。该实验说明了通过集成 LDA 主题模型的 ALS 算法效果要优于其他推荐算法。

关键词：推荐算法；矩阵分解；隐式狄利克雷分布；KL 散度；主题模型

中图分类号：TP301.6　　doi: 10.3969/j.issn.1001-3695.2017.12.0811

Recommendation algorithm combining theme model

Cao Zhanwei, Hu Xiaopeng

(1. *School of Information Science & Technology*, *Southwest Jiaotong University*, *Chengdu Sichuan* 611756, *China*)

Abstract: In order to solve the problem of cold start and data sparsity for traditional collaborative filtering recommendation algorithm, and the accuracy of similarity measurement, this paper proposed a matrix decomposition recommendation algorithm based on the LDA theme model. Firstly, it uses the improved LDA algorithm to output the project-topic distribution, using the perplexity as the modified function of the subject number; Secondly, it calculate the similarity matrix of the project based on the cosine similarity and the KL divergence, combineing the obtained similarity matrix with the original scoring training set to output the pre score, and then fills the preliminary score to the training set. Finally, it input the training set to ALS matrix decomposition algorithm to get the recommended results. The experimental results of the MovieLens data set show that the proposed algorithm can get a smaller MAE values than the traditional ALS algorithm under different implicit parameter settings and it greater than traditional recommdation algorithm . The experiment shows that the results of the ALS algorithm are better than other algorithms by integrating the LDA theme model.

Key words: recommendation algorithm; matrix decomposition; Latent Dirichlet distribution (LDA); KL divergence; theme model

0　引言

随着互联网的飞速发展，各大电商网站的数据呈现井喷式的增长，为满足用户在海量信息中进行有效选择的需求，推荐系统应运而生，而推荐算法[1]又是推荐系统的精髓。

主流推荐算法主要包括协同过滤推荐算法、基于内容的推荐、关联规则以及混合推荐方法。其中，协同过滤推荐算法[2]由于具有可利用用户行为数据和基于群体智慧的优势，在当前电商系统中应用最广泛。Zhengzheng 等人[3]提出了一种奇异值分解（singular value decomposition, SVD）的协同过滤算法，该法对高阶评分矩阵进行降维，缓解了数据稀疏的问题，但是由于计算复杂度过高以及存在冷启动问题，在商业领域运用并不广泛。Rahul 等[4]提出利用 K-Means 算法对用户进行聚类以减小邻居搜索空间,该法考虑到了用户对项目属性的偏好，推荐效果优于传统推荐算法，但是依然存在冷启动问题。Maryam 等人[5]针对用户兴趣的动态性，提出了 PIDFAR（potential interest discovery method based on fuzzy association rules）方法。该算法结合 LDA（latent dirichlet allocation），通过模糊关联规则挖掘出兴趣-时间模型，再根据关联规则和主题分布计算项目相似度。该算法在准确率上优于传统推荐算法并缓解了冷启动问题，但

收稿日期：2017-12-19；**修回日期：**2018-02-06

作者简介：曹占伟（1992-），男，四川达川人，硕士研究生，主要研究方向为推荐算法、云计算（530702649@qq.com）；胡晓鹏（1972-），男，陕西中人，副教授，博士，主要研究方向为软件架构、分布式云.

对数据稀疏问题未能深入讨论。

Zhou 等人[6]在 NetFlix 大赛中首次提出了基于交替最小二乘法（alternating least squares，ALS）的矩阵分解协同过滤算法。该方法在多用户、多项目以及稀疏数据的情况下优于经典的协同过滤算法，并在大赛中取得优异成绩，但是该法并未过多考虑新用户或新项目动态加入的因素，依然存在冷启动问题。

针对上述不足，本文提出了基于主题模型的 ALS 矩阵分解算法 LDA-IT-ALS（LDA insert to ALS ）。该方法运用 LDA 主题模型[7]将项目属性映射成输入文档，通过 LDA 算法输出项目之间的主题分布，进而得到相似度矩阵，然后通过此矩阵与原评分矩阵进行联合操作得到预评分，再将预评分填充到源矩阵中，最后通过 ALS 算法得到推荐结果。ALS 算法缓解了数据稀疏问题，本文算法在此基础上结合主题模型进行数据填充缓解了冷启动问题，并进一步缓解了数据稀疏问题，通过多次实验证明了本文方法能得到更低的预测误差。

1　ALS 矩阵分解算法与 LDA 主题模型

1.1　ALS 算法

协同过滤算法中用户对物品的评分可以表示成一个评分矩阵 R(m*n)，其中元素 R_{ij} 表示索引号为 i（0<i<=m）的用户对索引号为 j（0<j<=n）的物品的评分，如表 1 所示。

在推荐系统中用户对项目的评分往往低于 5%[8]，例如 MovieLens 数据集的稀疏度是 4.5%，Netflix 是 1.2% ，Bibsonomy 是 0.35%，Delicious 是 0.046%。因此表 1 所示的评分矩阵中的大多数元素往往为空。本文称这些空值为缺失值（Missing Value）。推荐系统中往往需要得到某用户对所有物品的评分，假设 R_{22} 为缺失值，则就需要通过某些方法预测 U_2 对 I_2 的评分，即“矩阵补全（填充）”。

ALS 矩阵补全即通过交替最小二乘法[9]来填补评分矩阵。ALS 算法的核心基于以下假设：评分矩阵 R 是近似低秩的，也就是说一个 m*n 的评分矩阵 R 可以用两个小稠密矩阵 X(m*k）和 Y(n*k)的乘积来近似表示，如表 2 和表 3 所示，其中 $R\approx XY^T$，k<<m、n，k 为隐式因子。

表 1　用户原评分矩阵

	I_1	I_2	I_3	…	I_n
U_1	R_{11}	R_{12}	R_{13}	…	R_{1n}
U_2	R_{21}	R_{22}	R_{23}	…	R_{2n}
U_3	R_{31}	R_{32}	R_{33}	…	R_{3n}
…	…	…	…	…	…
U_m	R_{m1}	R_{m2}	R_{m3}	…	R_{mn}

表 2　稠密矩阵 X

	F_1	F_2	F_3	…	F_k
U_1	X_{11}	X_{12}	X_{13}	…	X_{1k}
U_2	X_{21}	X_{22}	X_{23}	…	X_{2k}
U_3	X_{31}	X_{32}	X_{33}	…	X_{3k}
…	…	…	…	…	…
U_m	X_{m1}	X_{m2}	X_{m3}	…	X_{mk}

表 3　稠密矩阵 Y

	F_1	F_2	F_3	…	F_k
I_1	Y_{11}	Y_{12}	Y_{13}	…	Y_{1k}
I_2	Y_{21}	Y_{22}	Y_{23}	…	Y_{2k}
I_3	Y_{31}	Y_{32}	Y_{33}	…	Y_{3k}
…	…	…	…	…	…
I_n	Y_{n1}	Y_{n2}	Y_{n3}	…	Y_{nk}

ALS 矩阵分解的目的是将 User 和 Item 映射到一个维度为 k（k<<m、n）的隐式空间，这样 User 对 Item 的评分就可以通过隐式空间矩阵建模。ALS 算法求解矩阵 X 和 Y 方法如算法 1。

算法 1　ALS 矩阵分解算法

a）定义损失函数 $C=\sum_{(i,j)\in R}\left[\left(r_{i,j}-x_i^T y_j\right)^2+\lambda\left(\|x_i\|^2+\|y_j\|^2\right)\right]$，其中 $r_{i,j}$ 代表初始评分矩阵中用户 i 对项目 j 的评分，x_i 为 X（m*k）的第 i 行的一个列向量，y_j 为 Y（n*k）的第 j 行的一个列向量，λ 为正则化参数；

b）随机生成一个 X(0),一般可以取 0 值或者全局均值；

c）固定 X(0),即将 X(0)当作常量，求解 Y(0)；此时的损失函数为 $C=\sum_{(i,j)\in R}\left[\left(r_{i,j}-\left(x_i^{(0)}\right)^T y_j\right)^2+\lambda\left(\left\|x_i^{(0)}\right\|^2+\|y_j\|^2\right)\right]$

d）由于 C 中只有 V_j 一个未知变量，因此 C 的最优化问题转换为最小二乘问题，即用最小二乘法求解 Y_j 的最优解。固定 j（j=1,2,......,n），则 C 的导数

$$\frac{\partial C}{\partial y_j}=\frac{\partial}{\partial y_j}\left[\sum_{i=1}^{m}\left[\left(r_{i,j}-\left(x_i^{(0)}\right)^T y_j\right)^2+\lambda\left(\left\|x_i^{(0)}\right\|^2+\|y_j\|^2\right)\right]\right]=$$

$$\sum_{i=1}^{m}\left[2\left(r_{i,j}-\left(x_i^{(0)}\right)^T y_j\right)\left(-x_i^{(0)}\right)+2\lambda y_j\right]$$

$$=2\sum_{i=1}^{m}\left[\left(\left(x_i^{(0)}\right)^T x_i^{(0)}+\lambda\right)y_j-r_{i,j}y_i^{(0)}\right]$$

e）令 $\frac{\partial C}{\partial v_j}=0$ ，可得 $\sum_{i=1}^{m}\left[\left(\left(x_i^{(0)}\right)^T x_i^{(0)}+\lambda\right)y_j\right]=\sum_{i=1}^{m}r_{i,j}x_i^{(0)}$ $\left(XX^T+\lambda E\right)y_j=Xr_j^T$ 。

f）令 $M_1=XX^T+\lambda E, M_2=Xr_j^T$ ，可得 $y_j=M_1^{-1}M_2$ ；

g)按照步骤 c)~f))依次计算 $y_1, y_2, y_3,\dots, y_n$,从而得到 Y(0)；

h）固定 Y(0)，来求解 X(1)。求解方法同步骤 g)，得到 $\left(YY^T+\lambda E\right)x_i=Yr_i^T$

令 $M_1=YY^T+\lambda E, M_2=Yr_i^T$ 得 $x_i=M_1^{-1}M_2$ 。

i）依照步骤 h)依次计算 $x_1, x_2, x_3,\dots, x_m$,从而得到 X(1)。

j）循环交替执行步骤 a)~i),直到损失函数 C 的值收敛或者达到设定的迭代次数，这样就得到了最优解对应的矩阵 X，Y。

由算法 1 得到的矩阵 X 和 Y 即可用来近似求解稠密的评分矩阵 $R'=XY^T$， R'中相较于 R 多出的评分项即为预评分项，

可据此对用户做推荐。

1.2 LDA 主题模型

1.2.1 主题模型概念

主题模型（Topic Model）是用来在大量文档中提取抽象主题的一种统计模型，它可以根据不同文档中出现相同或者不同词汇的条件概率确定文档的隐主题以及词汇的主题归属。由于同一个主题可能会包含多个词汇，因此文档的比较不再是简单的词汇统计，而是文档中主题出现的概率分布。某个主题中包含的词汇是语义相对相似或者相同的，因此主题可以表示为整个文档集词汇表中词汇的多元分布函数。词汇对应的系数越大，那么词汇的语义越接近主题的语义；反之，和主题的关系越小。归纳起来，文档就是不同主题对词汇集不同词汇分布的分布。

由于 TF-IDF 方法对文本主题分析的局限性，先后出现了潜在语义分析模型（latent semantic analysis，LSA）、pLSA 模型以及马尔科夫模型等，但多存在计算复杂度高、针对多文档的过拟合以及扩展性不好等问题，但它们的出现为潜在狄利克雷分布 LDA 主题模型的出现奠定了基础[10]。

1.2.2 LDA 算法描述

Blei 等人[7]在 2003 年提出的潜在狄利克雷分布（LDA）是一种无监督的生成模型。该方法认为每篇文档的生成方式如下：先从主题集合中以一定的概率抽取主题，然后再从这个主题对应的词汇集合中以一定的概率抽取当前词汇，循环执行，直到生成整篇文档。

一个词汇的生成概率如式（1）所示。

$$p(\text{word}\mid\text{doc})=\sum_{topic}p(\text{word}\mid\text{topic})\times p(\text{topic}\mid\text{doc}) \tag{1}$$

LDA 模型的生成过程如图 1 所示，其中 K 为主题个数，M 为总文档数，Nm 是第 m 个文档的词汇总数，β 是每个 Topic 下词的多项分布的 Dirichlet 先验超参数，α 是每个文档下 Topic 的多项分布的 Dirichlet 先验超参数，$Z_{m,n}$ 是第 m 个文档中第 n 个词汇所属主题，$w_{m,n}$ 是第 m 篇文档中的第 n 个词汇，两个隐变量 θ_m 和 ϕ_k 分别表示第 m 篇文档下的 Topic 分布和第 k 个 Topic 下 word 的分布，前者是 k 维(k 为 Topic 总数)向量，后者是 v 维向量，其中 v 为所有文档集中不重复词汇的总数。

图 1 所示的文档生成过程主要分解为两个物理过程：

a）$\vec{\alpha}\to\vec{\theta}_m\to z_{m,n}$，表示在生成第 m 篇文档的时候，先从 M 个 doc-Topic 分布函数中抽取分布函数 $\vec{\theta}_m$，然后从多个 Topic 中抽取一个编号为 $z_{m,n}$ 的 Topic 作为第 n 个词的主题；

b）$\vec{\beta}\to\vec{\varphi}_k\to w_{m,n}\mid k=z_{m,n}$，表示在 K 个 Topic-word 分布函数中选择编号为 $k=z_{m,n}$ 的分布，然后在分布中随机选择 word 作为第 m 篇文档的第 n 个词 $w_{m,n}$；

在给定超参数 α 和 β 的情况下，LDA 的联合共轭分布如式（2）所示，其中 $\vec{n_k}=(n_k^{(1)}\ldots n_k^{(V)})$，$n_k^{(t)}$ 表示第 m 篇文档中第 k 个主题产生词汇的个数，k 下标为文档编号。$\vec{n_m}=(n_m^{(1)}\ldots n_m^{(K)})$，$n_m^{(t)}$ 表示第 m 篇文档中第 k 个主题产生的词汇的个数，k 下标为 Topic 编号。

$$\begin{aligned}p(\vec{w},\vec{z}\mid\vec{\alpha},\vec{\beta})&=p(\vec{w}\mid\vec{z},\vec{\beta})p(\vec{z}\mid\vec{\alpha})\\&=\prod_{k=1}^{K}\frac{\Delta(\vec{n_k}+\vec{\beta})}{\Delta\vec{\beta}}\prod_{m=1}^{M}\frac{\Delta(\vec{n_m}+\vec{\alpha})}{\Delta\vec{\alpha}}\end{aligned} \tag{2}$$

以上是 LDA 定义的文档生成过程，而实际情况是 LDA 生成过程的逆过程。已知文档集，需要对联合分布 $p(\vec{w},\vec{z})$ 进行采样。采样的方法包括变分-EM 算法、Gibbs 抽样法和最大似然估计法。本文采用 Gibbs 抽样法采样主题分布，有关细节可查看文献[11]。

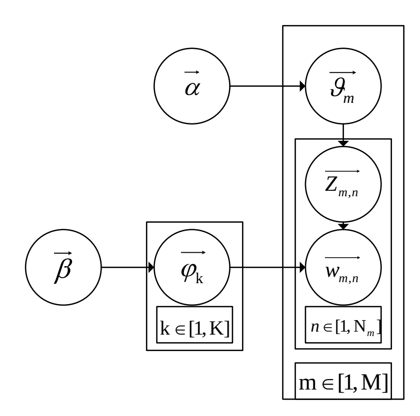


图 1 LDA 图模型

2 本文改进算法

ALS 矩阵分解推荐算法虽然优于基于邻域的协同过滤算法，但是依然存在数据稀疏以及冷启动问题，这些问题对实际推荐准确率有很大影响。本文提出了一种结合 LDA 主题模型的 ALS 算法，即 LDA-IT-ALS 算法。该算法利用 LDA 模型对项目信息进行建模，将项目文件中的每一行（单个项目的描述信息）转换为主题的分布，不仅对文档进行了降维而且挖掘出了不同词可能包含的相同或相近的主题信息。在这些信息的基础上进行的项目相似度的计算更符合人的常规思维方式。主题分布信息经过本文方法处理后变成项目相似度矩阵，然后结合原评分矩阵产生预评分填充到原评分矩阵中形成新的输入集输入 ALS 算法。

2.1 算法过程描述

本文所用到的数据集为美国 GroupLens 提供的 MovieLens 数据集[12]。算法主要包括项目文件处理及主题分布的计算、项目相似度矩阵求解、预评分求解及填充、ALS 矩阵分解及推荐。

2.1.1 文件处理及主体分部计算

本步骤用到了 u.Item 文件以及 u.genre 文件。文件 u.Item 每行描述一部电影，不同的描述项（电影属性）以单竖线分割：第 1 项为索引号，第 2 项为电影名、第 3 项为上映日期，第 4 项为空，第 5 项为网址信息，第 6~24 项为以位图描述的电影类型（如果某部电影属于某个类型，则对应的位图为 1，否则为 0）。文件 u.genre 描述电影类型及索引号，每行描述一个电影类型，共 19 行。

本步骤又可分为两个子过程，处理流程如图 2 所示。

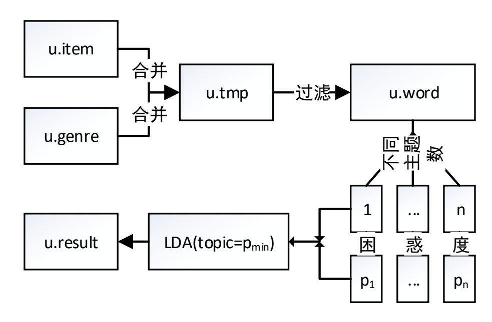


图 2　项目文件处理

1）电影描述文件预处理

u.Item 文件不能直接作为主题分布计算模块的输入文件。需要将 u.Item 的位图信息转换为电影类型信息，提取电影名称及上映日期，并过滤掉干扰信息。经过处理后的文件为 u.word，文件中每一行表示一部电影的文本信息，例如电影 Toy Story 的信息为（Toy Story Jan1995 Animation Children's Comedy）。

2）主题分布计算

将 u.word 转换成主题分布文件，用到了 LDA 算法。由于 Blei 提出 LDA 算法的初衷是利用隐式主题对文本进行分类，源码在读文件模块读入对象是以一个项目对应一个文件的形式存在，而步骤 1）中得到的 u.word 文件约 1600 行文本，其中一行代表一部电影。为满足后续实验要求以及避免大量文件读取时造成频繁的磁盘读操作而降低效率，本文对 Blei 的读模块进行了改进，将逐文件处理变为了逐行处理。

在完成读模块的改写后，需要对 u.word 进行联合分布的采样以及主题分布的输出，具体过程参考 1.2.2 节。为提高主题分布对电影的区分效果，本文引入了困惑度[13]，表示对不同物品间的区分能力，困惑度越小其区分效果越好。困惑度由式（3）所示，其中 W 为语料集，w_m 为语料中的单词，N_m 为单词数量。

$$perplexity(W)=\exp\left\{-\frac{\sum_{m}\ln p\left(w_m\right)}{\sum_{m}N_m}\right\} \tag{3}$$

在不同的主题数前提下得到多个具有不同困惑度的主题分布，设置最小困惑度对应主题数进行再一次 LDA 算法，得到的主题分布矩阵如图 3 所示，其中 $p_{ij}(1\le i\le m,1\le j\le n)$为第 i 个项目的第 j 个主题的条件概率。

$$\begin{pmatrix} p_{11} & p_{12}\cdots & p_{1n} \\ p_{21} & p_{22}\cdots & p_{23} \\ \vdots & \vdots & \vdots \\ p_{m1} & p_{m2}\cdots & p_{mn} \end{pmatrix}$$

图 3　主题分布矩阵

2.1.2 项目相似度矩阵求解

在主题分布矩阵基础上进行项目相似度矩阵求解，主要过程为两两之间计算相似度。

1）相似度选择

常见的相似度的度量方式有余弦相似度、Jaccard、欧式距离等，最为常用的为余弦相似度。本文对余弦相似度和文献[14]提到的 KL 散度进行对比。

余弦相似度是通过计算两个向量的夹角余弦值来评估它们的相似度，计算公式如式（4）所示，其中 A_i 代表向量 A 的在维度 i 上的值，B_i 代表向量 B 的在维度 i 上的值，n 为维度。

KL 散度又叫相对熵或者互熵、交叉熵，用于度量两个随机变量的距离，计算公式如式（5）所示，其中 p(x)和 q(x)是关于变量 X 取值的概率分布函数。由于 KL 散度是两个概率分布 P 和 Q 的非对称性的度量，一般采用(D(p||q)+D(q||p))/2 度量两个分布间的距离。在此需要将距离转换为相似度，并且归一化。用常用的 sigmoid 函数进行转换，公式如式（6）所示，其中 D 为(D(p||q)+D(q||p))/2。

$$\cos\theta=\frac{\sum_1^n(A_i\times B_i)}{\sqrt{\sum_1^n A_i^2}\times\sqrt{\sum_1^n B_i^2}} \tag{4}$$

$$\mathrm{D}(p\,\|\,q)=\sum_1^n p(x)\log\frac{p(x)}{q(x)} \tag{5}$$

$$S(\mathrm{x})=\frac{1}{1+e^{-D}} \tag{6}$$

2）相似度后处理

将图 3 中的主题分布矩阵分别根据 1）中的两种方式逐行两两计算相似度，得到相似度矩阵。为方便后续流程，生成的相似度矩阵实际上以三元组形式存在，形如（item1，item2，sim）。

后续求解预评分根据邻居项目已存在的评分来估计空白处的评分，因此相似度矩阵需要以固定阈值过滤掉相似度小的项。本文按照文献[16]的方法取阈值为 0.9，得到的相似度矩阵后续称为高相似度矩阵。

2.1.3 预评分求解及填充

结合原评分矩阵以及高相似度矩阵预测某些缺失评分项，原评分矩阵文件为 u.data，其评分是以三元组形式存在，形如（user，item，rate），本文主要根据当前项目和邻居项目间的相似性集合和用户对项目结合的评分预测缺失项，主要过程如算法 2。

算法 2　预评分求解

a)定义 preMarkMap ，其数据结构为 Map<Integer,Map<Integer,Double>>

b)定义 markMap 存储评分数据，其数据结构为 Map<Integer,Map<Integer,Double>>

c)逐行遍历原评分文件，User 和 ItemMap 保存在 markMap 中，形如$[(u_1,[(i_1,v_{11}),(i_2,v_{12})]),(u_2,[(i_1,v_{21}),(i_2,v_{22})])\ldots]$

d)定义 ItemSimMap，数据结构为 Map<Integer,Map<Integer,Double>>

e)逐行遍历相似度矩阵，Item1 和 Item2List 保存于 ItemSimMap 中，形如$[(i_1,[(i_2,v_{12}),(i_3,v_{13})]),(i_2,[(i_3,v_{23}),(i_4,v_{24})])\ldots]$

f)遍历 markMap，获取当前 User 的 ItemMap，遍历每个

ItemSimMap，如果 ItemMap 中不包含当前 Item 值，那么当前 User 对当前 Item 的预评分可通过如下方式求解：

(a)通过当前 Item 获取其 value1Map 形如[(i_2,v_{12}),(i_3,v_{13})…]

(b)通过当前 User 获取其 value2Map 形如[(i_1,v_{11}),(i_2,v_{12})…]

(c)设置局部变量 sum 和 n

(d)遍历 value1Map，获取当前 ItemTmp，当前值为 value1Tmp

.遍历 value2Map 查找键为 ItemTmp 的项，值为 value2Tmp

. 将 (value1Tmp*value2Tmp+value1Tmp/value2Tmp)/2 叠加到 sum，n 自增 1

(e)将当前 User，ItemTmp，sum/n 的值写入 preMarkMap

g）循环执行步骤 e）f)，最后输出 preMarkMap。

算法 2 简要的描述了预评分求解过程，将输出的预评分填充到原评分文件即可作为 ALS 矩阵分解算法的训练集。

2.1.4 矩阵分解及推荐

在得到训练集的前提下，利用 1.1 节的算法进行求解并推荐。在 1.1 节提到了隐式参数，本算法将平均绝对误差 MAE[15] 值作为隐式参数的修正目标函数：MAE 越小证明推荐预测效果越好。MAE 定义如式（7），其中 p_i 为预测评分，q_i 为真实评分，N 为测试集评分数。

$$MAE = \frac{\sum_{i=1}^{N} |p_i - q_i|}{N} \tag{7}$$

通过修正 MAE 得到的最优模型即可对用户进行推荐，比如对用户 U_1 的预测矩阵如表 3 所示，那么就可以根据预测集合优先推荐 I_3 和 I_6 给用户 U_1。

表 3 用户预测评分矩阵

	I_1	I_2	I_3	…	I_6
U_1	3.3	3.5	4.6	…	5.7

3 实验对比及结果分析

本实验使用的数据集来自美国 GroupLens 研究小组提供的 MovieLens 数据集。数据集中包括 943 个用户描述文件，1682 个电影信息描述文件，10 万个评分记录。本文从数据集中抽取 80%作为训练集，利用 LDAIT-ALS 算法进行推荐，20%的测试集检测算法的效果。使用 5 折交叉平均实验结果减少误差，采用 MAE 作为评价标准，实验方法如第 2 节所示。本文算法由 JAVA 实现，实验平台为 Windows7-64 位双核，IDE 为 Eclipse。

3.1 与 ALS 算法对比实验

3.1.1 LDA 主题数设置

为了能保证 LDA 算法的处理效果，首先需要对主题数的设置进行调整，以困惑度作为主题数的修正参数。

首先将主题数设置在 5~50，步长为 5，困惑度与主题数的关系如图 3 所示，topNum 代表主题数，perplexity 代表困惑度，发现在 topNum 为 5-10 之间 perplexity 最小。然后设置 topNum 在 1-10，步长为 1，困惑度与主题数的关系如图 4 所示。困惑度在 topNum 为 6 的时候最小，因此本文后续实验主题数固定为 6。

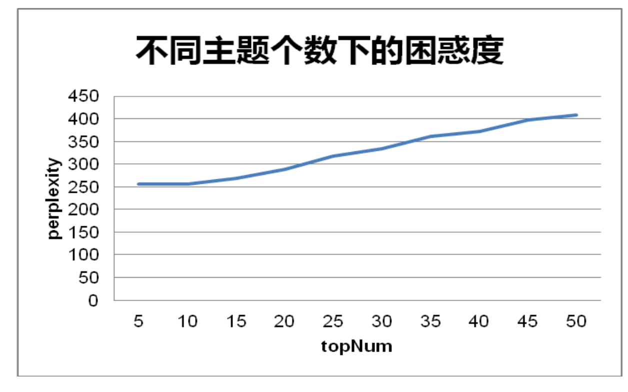


图 3 困惑度与主题数(步长为 5)

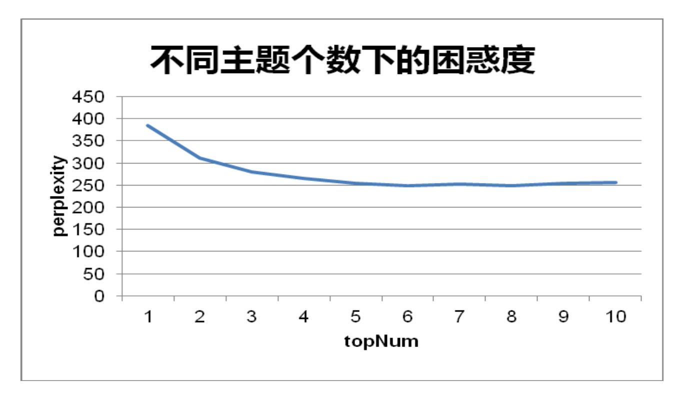


图 4 困惑度与主题数(步长为 1)

3.1.2 与 ALS 算法对比

为验证本文算法的有效性，在不同隐式参数设置下将 ALS 算法和本文算法进行对比。另外，还将本文算法中处理项目相似度的方式进行纵向对比。

首先设定隐式参数从 5~40，步长为 5，对比结果如图 5 所示。发现在 5~10 的 MAE 值最小，然后设定隐式参数从 5~10，步长为 1 进行对比。结果如图 6 所示，在 MAE 为 3 的时候三种算法均能达到最小误差，并且本文算法的 MAE 值小于 ALS 算法的 MAE 值。

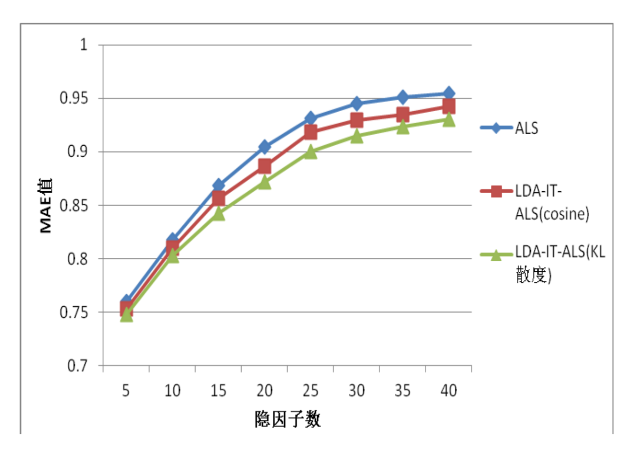


图 5 隐因子 step=5 时算法对比

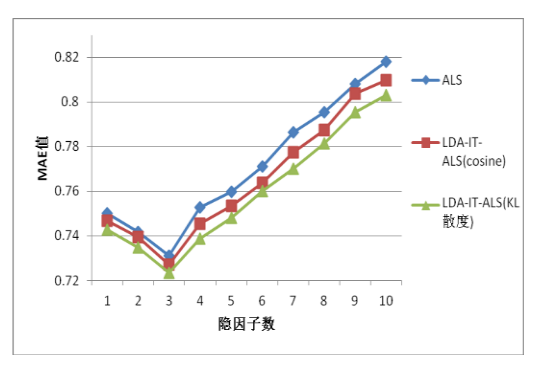


图 6　隐因子 step=1 时算法对比

3.2　其他推荐算法比较

为了进一步证明本文算法的有效性，择取文献[17]提出的算法以及基于用户和基于物品的协同过滤算法进行比较。由于后三种算法采用的是邻居用户或者邻近项目的调参方式，不同于本文，所以采取最优值进行比较。为保证实验结果的有效性，采用统一数据集，统一实验平台以及统一评价标准 MAE，将各个算法调参后的最优值作为最后统计结果。调参方法分别参考本文 3.1 节以及文献[17]，最后得到的结果如图 7 所示。

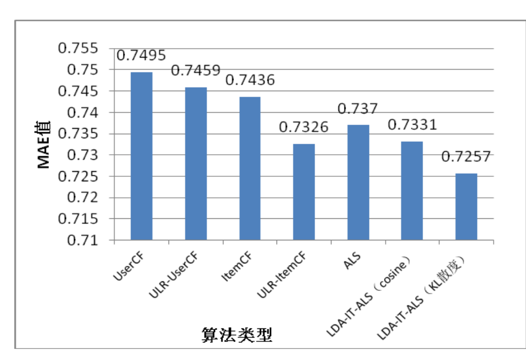


图 7　推荐算法对比

3.3　实验结果分析

通过实验证明，本文算法平均绝对误差在设定不同隐因子的情况下均小于 ALS 算法，同时也要优于文献[17]的算法。通过图 6 可以发现 ULR-ItemCF 算法优于 ItemCF 算法和 ALS 算法，原因在于 ULR-ItemCF 算法也是利用了项目的描述信息，缓解了冷启动问题，这也进一步证明了项目描述信息在推荐算法中的重要性。同时，也可以发现本文算法 LDA-IT-ALS 的最小 MAE 值为 0.7257，效果要好于 ULR-ItemCF 算法，原因在于本文算法采用的主题模型生成的预评分正好填充了 ALS 算法训练集所缺的评分项，缓解了数据稀疏性并且解决了冷启动问题。例如，若表 1 中 I_3 是新上映电影，那么 R_{13}~R_{m3} 都为空值，如果将此时数据集作为训练集，推荐系统无法对用户进行个性化推荐，也即出现了冷启动问题。在利用本文方法预测空白值并填充后，再根据后续步骤即可进行正常推荐。另外，在实验中对项目相似度矩阵求解方法余弦相似度和 KL 散度进行对比，结果 KL 散度实验效果优于余弦相似度。原因是 KL 散度本身更适合计算两个概率分布之间的距离，而余弦相似度更适用于实际空间物理角度的计算。

4　结束语

本文提出的结合 LDA 主题模型的 ALS 推荐算法（LDA-IT-ALS）属于管道式[14]的集成算法。利用改进算法建模将项目描述信息处理成主题分布信息，然后融入到评分文件中，解决了冷启动问题并缓解了数据稀疏问题，使推荐准确度得到提高。再者，本文对 LDA 原创作者的源码进行了改进以适应本实验的处理流程，避免了磁盘频繁读取的问题。另外，在项目相似度计算时引入了 KL 散度，提升了相似度度量准确率，减小了推荐误差，与传统推荐算法相比准确度有所提高。但是推荐算法依然面临许多问题，比如安全性、实时性问题。本文后续工作将着重于将 LDA-IT-ALS 算法并行化，实现其在线实时处理。

参考文献：

[1] Guo Yan, Wang Minxi, Li Xin. Application of an improved Apriori algorithm in a mobile e-commerce recommendation system [J]. Industrial Management & Data Systems, 2017, 117 (2) .

[2] Wei Jian, He Jianhua, Chen Kai, *et al.* Collaborative filtering and deep learning based recommendation system for cold start items [J]. Expert Systems with Applications, 2017, 69.

[3] Xian Zhengzheng, Li Qiliang, Li Gai, *et al.* New collaborative filtering algorithms based on SVD+and differential privacy [J]. Mathematical Problems in Engineering, 2017, 2017.

[4] Katarya R, Verma O P. An effective collaborative movie recommender system with cuckoo search [J]. Egyptian Informatics Journal, 2016.

[5] Najafabadi M K, Mahrin M N, Chuprat S, *et al.* Improving the accuracy of collaborative filtering recommendations using clustering and association rules mining on implicit data [J]. Computers in Human Behavior, 2017, 67 (C): 113-128.

[6] Zhou Y, Wilkinson D, Schreiber R, *et al.* Large-Scale Parallel Collaborative Filtering for the Netflix Prize [C]// Proc of International Conference Algorithmic Aspects in Information and Management. 2008: 337-348.

[7] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation [J]. Journal of Machine Learning Research, 2003, 3: 993-1022.

[8] Villegas N M, Sánchez C, Díaz-Cely J, *et al.* Characterizing context-aware recommender systems: a systematic literature review [J]. Knowledge-Based Systems, 2017.

[9] 印鉴，王智圣，李琪，等. 基于大规模隐式反馈的个性化推荐 [J]. 软件学报, 2014, 25 (9): 1953-1966.

[10] Peritz M. Ein Bruchstück aus J'hûdah Hajjûġ's arabischem Werke über die hebräischen Zeitwörter mit schwachen Stammlauten [J]. Zeitschrift für die Alttestamentliche Wissenschaft, 2009, 13 (1) .

[11] Chai Huimin, Lei Jiangnan, Fang Min. Estimating Bayesian networks parameters using EM and Gibbs sampling [J]. Procedia Computer Science, 2017, 111: 160-166.

[12] MovieLens 100K Dataset [DB/OL]. https://grouplens.org/datasets/

movielens/

[13] Zhao Weizhong, Chen J J, Perkins R, *et al.* A novel procedure on next generation sequencing data analysis using text mining algorithm [J]. BMC Bioinformatics, 2016, 17 (1): 301.

[14] 项亮. 推荐系统实践 [M]. 北京: 人民邮电出版社, 2012.

[15] Zhao Weizhong, Chen J J, Perkins R, *et al.* A heuristic approach to determine an appropriate number of topics in topic modeling [J]. BMC Bioinformatics, 2015, 16 (13): S8

[16] Moreno M N, Segrera S, López V F, *et al.* Web mining based framework for solving usual problems in recommender systems: a case study for movies′ recommendation [J]. Neurocomputing, 2016, 176 (C): 72-80.

[17] 高娜, 杨明. 嵌入LDA主题模型的协同过滤推荐算法 [J]. 计算机科学, 2016, 43 (3): 57-61, 79.